

Image Dataset Curation

Workshop IV

Antonio Rueda-Toicen
AI Engineer

KI Service Zentrum, Hasso Plattner Institute
February 2024

KI Service
Zentrum
by Hasso-Plattner-Institut

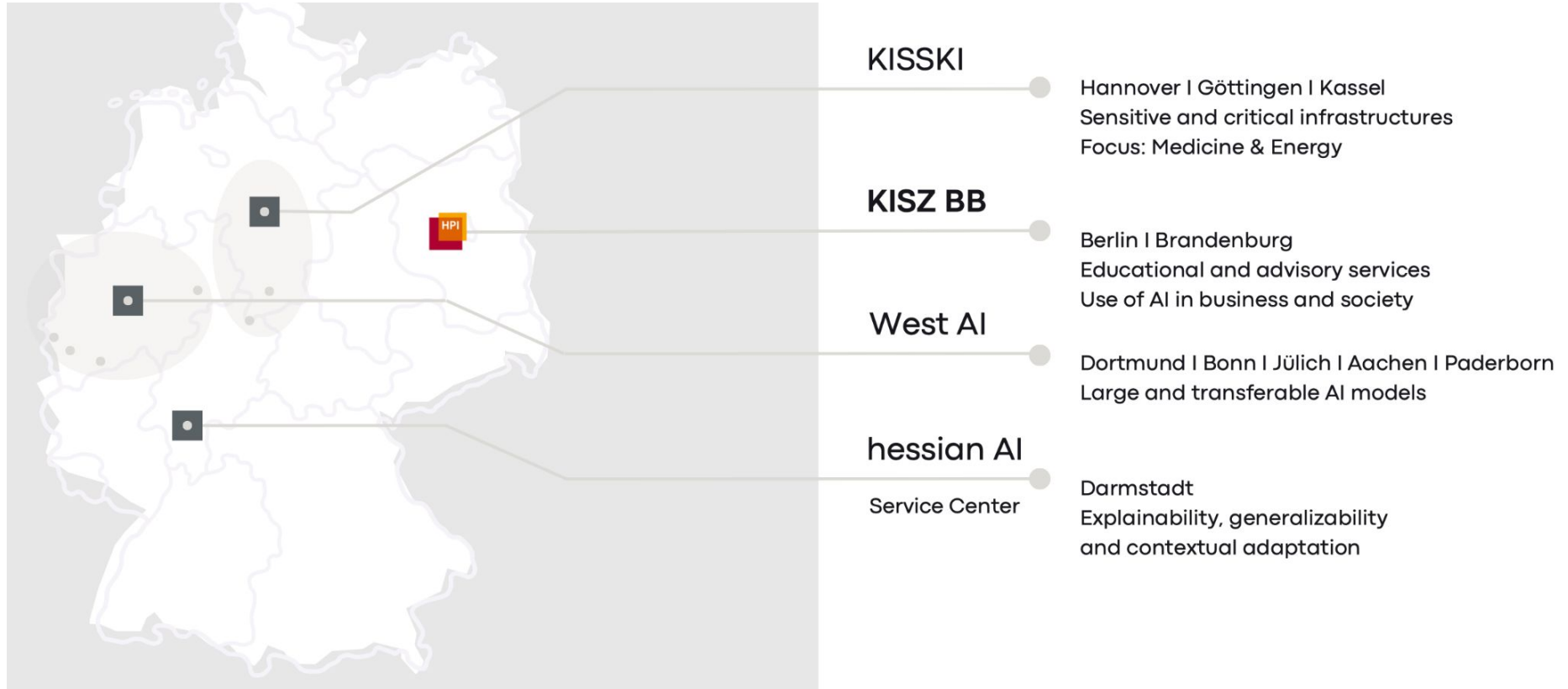


SPONSORED BY THE



Federal Ministry
of Education
and Research

<https://hpi.de/kisz/home.html>



<https://hpi.de/kisz/home.html>

Agenda

- Brief intro and learning objectives
- Overview of parts I and II of this workshop series
 - Understanding image embeddings
 - Scraping images from Google Images
- Classifying images with pretrained Resnets
- Multi-label vs single label classification
- Fine-tuning a Resnet with FastAI
- Review questions and discussion

What we expect you to have

- Some Python knowledge
- Curiosity :)

Learning objectives

At the end of the first workshop you will be able to:

- Describe use cases for image similarity in dataset curation
- Scrape images from Google Images or Bing
- Generate embeddings for images using a pretrained neural network
- Compare image pairs using cosine similarity
- Visualize embeddings in 3D using Tensorboard

Learning objectives

At the end of the second workshop you will be able to:

- Visualize image neighborhoods with k-nn
- Cluster images using k-medoids
- Select representative images

Learning objectives

At the end of the third workshop you will be able to:

- Classify images with a pre-trained Resnet
- Fine-tune a Resnet for custom classes

Learning objectives

At the end of the fourth workshop you will be able to:

- Explore class-activation mappings (CAM)
- Classify images with CLIP
- Clean an image dataset with Cleanlab

How are we doing this workshop

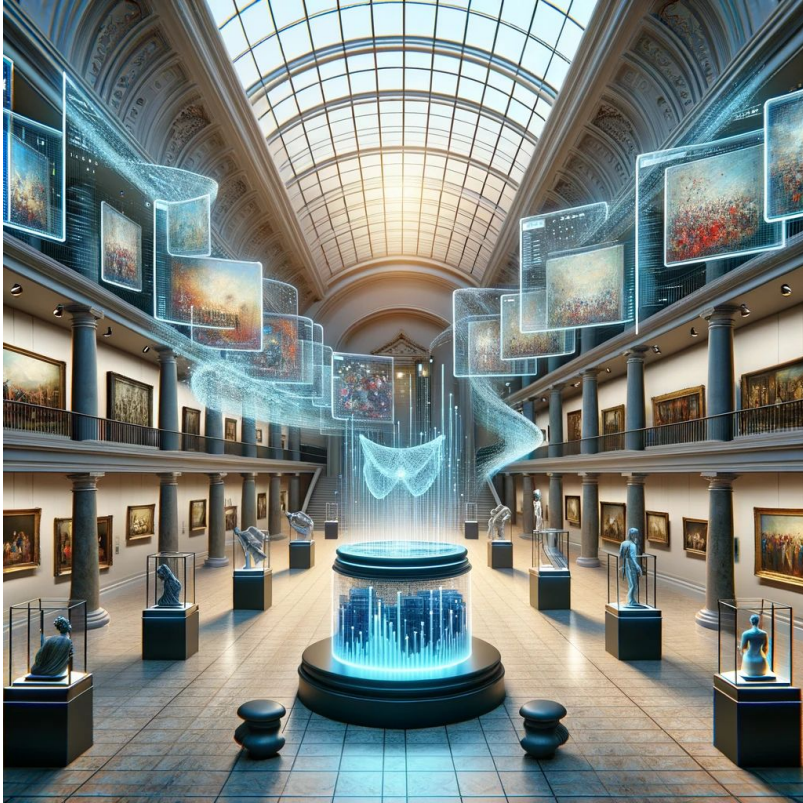
- We **type** most of the Python code from the tutorial notebooks in Google Colab

Repository: <https://github.com/KISZ-BB/image-dataset-curation-workshops>

What you need

- A Google user account
- A Google Drive account with enough free space
- Google Chrome or Firefox

What is dataset curation?



- We want to make data **accurate** and **relevant**
- We clean, deduplicate, and label
- This is similar to what museum curators do

Quick exploration of the dataset using embeddings

[Exploring the lions dataset with embeddings](#) (Colab notebook)

1.0



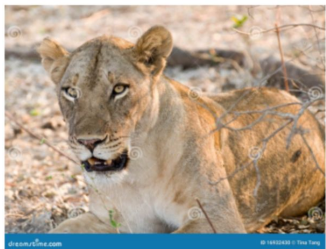
0.91



0.91



0.9



0.9



0.9



Lion or not a lion?

“If you torture the data long enough, it will confess to anything” Ronald Coase, economist



- [Was it a lion?](#)

Class Activation Mapping (CAM)

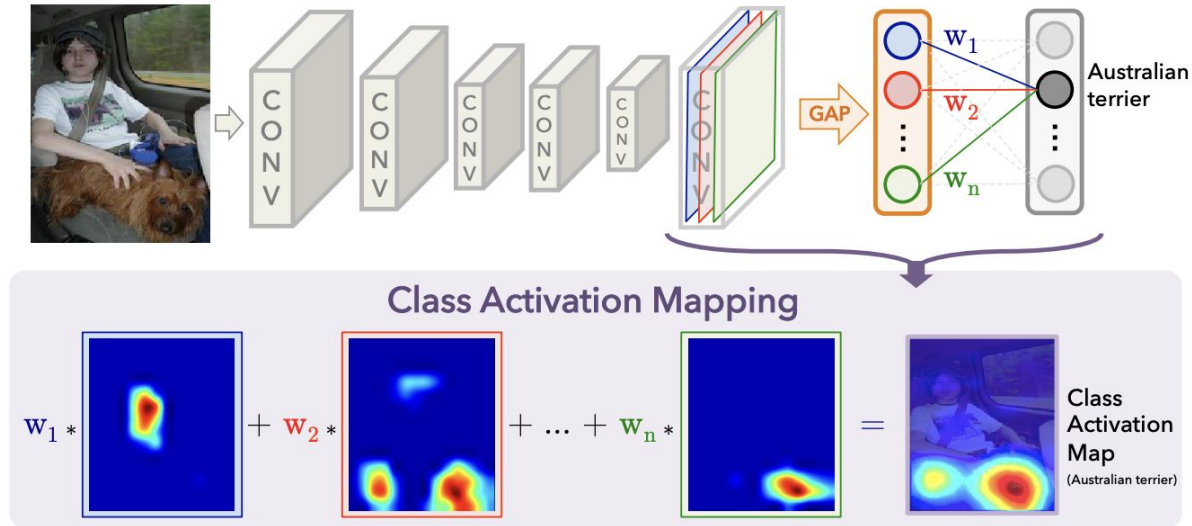
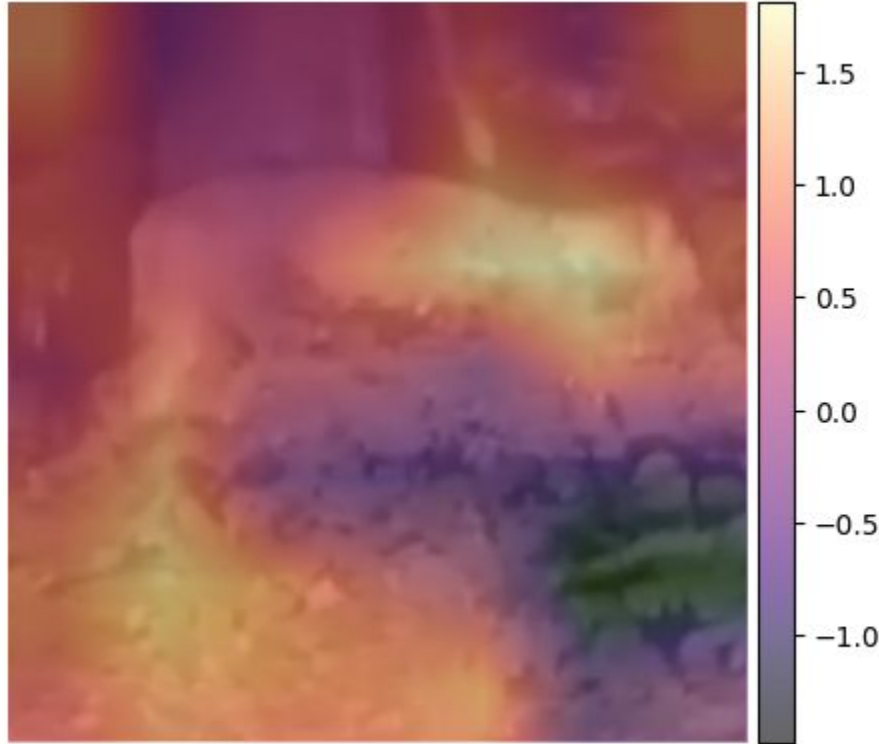


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

Spurious correlations come from the training data



Here the model associates the **front of the animal** and the **vegetation** with the classes 'boar' ($p=0.53$) and 'male lion' ($p=0.66$)

[Explore the notebook](#)

- Which classes do you get?
- How does the CAM map look like?



T+ 23 Löwen in Brandenburg gemeldet Haltung von Großkatzen – das ist die Rechtslage in Berlin und der Region

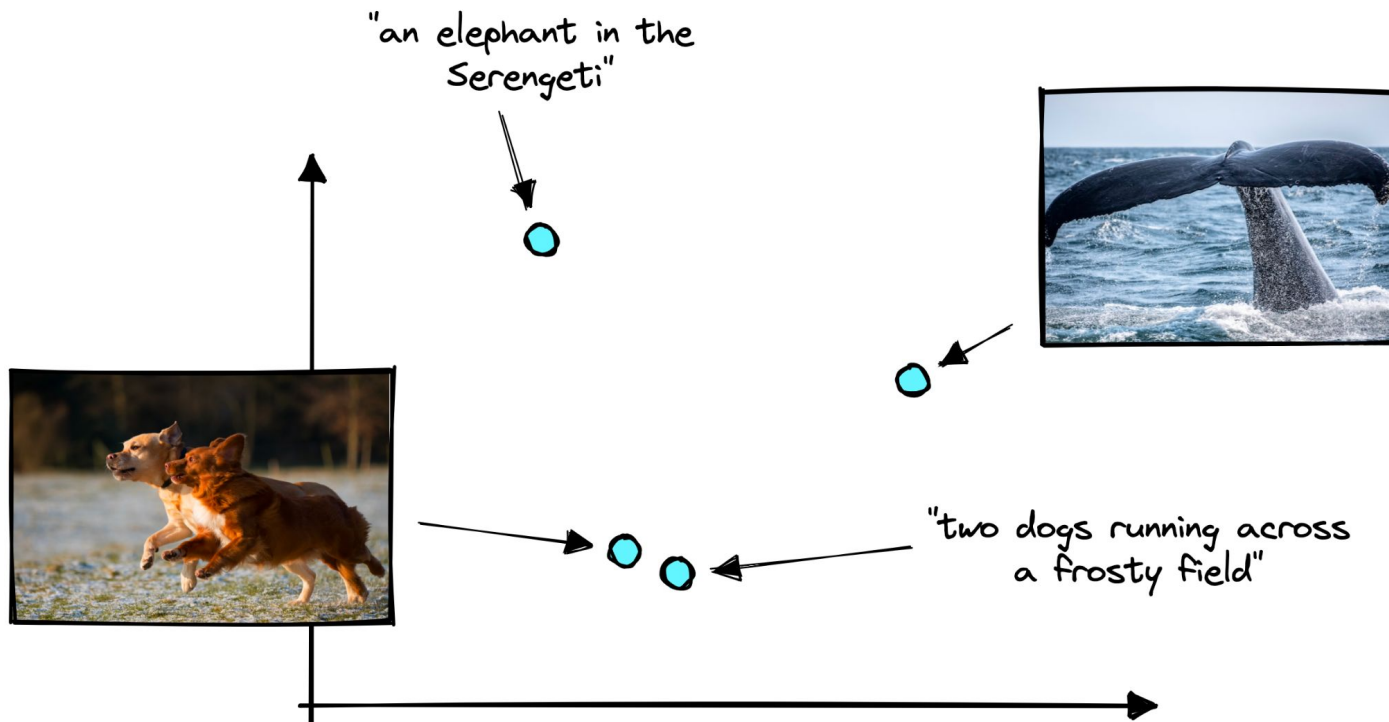
Anders als in Brandenburg ist die private Wildtierhaltung in Berlin – mit einigen Ausnahmen – verboten. Wer es trotzdem tut, muss mit einer hohen Strafe rechnen.

Von [Alexander Fröhlich](#) und [Daniel Böldt](#)

21.07.2023, 11:43 Uhr

[23 privately owned lions in Brandenburg!](#)

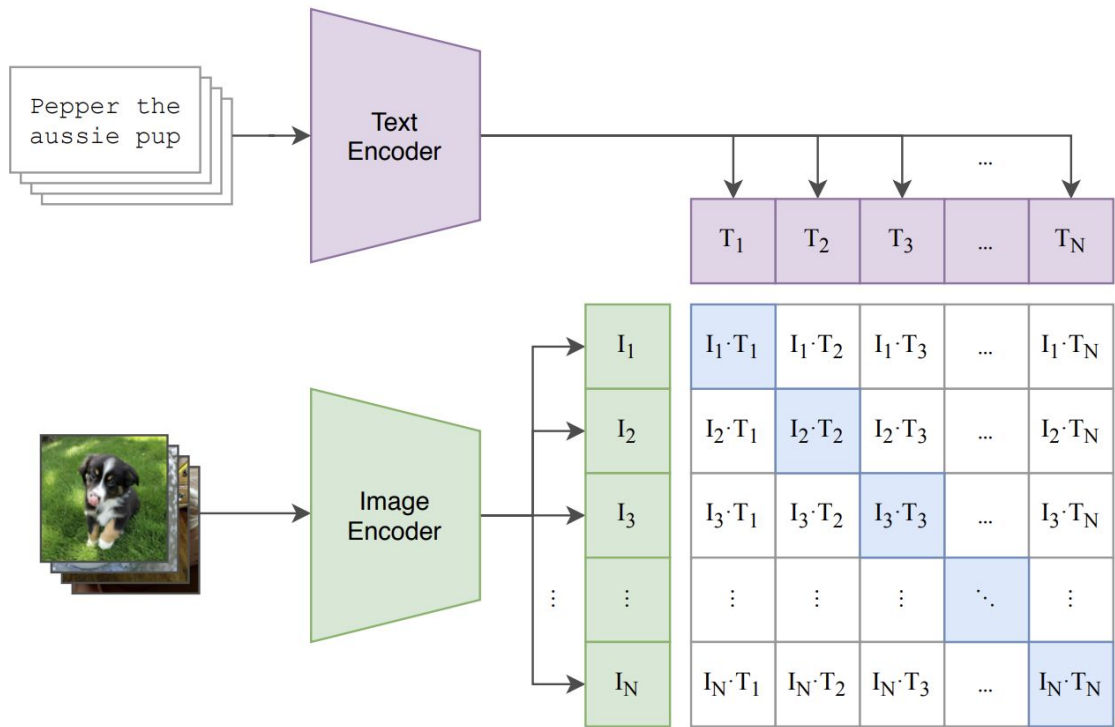
OpenAI's CLIP



[Image from pinecone's blogpost](#)

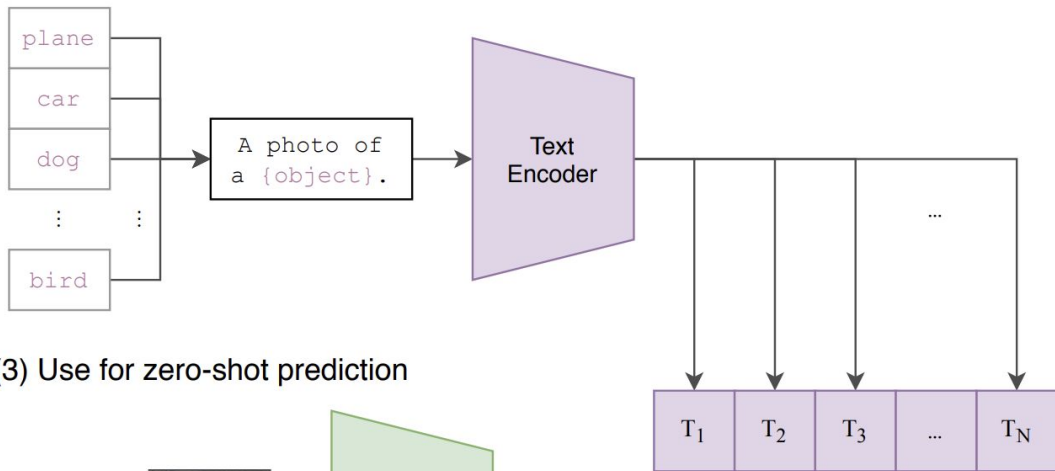
OpenAI's CLIP

(1) Contrastive pre-training

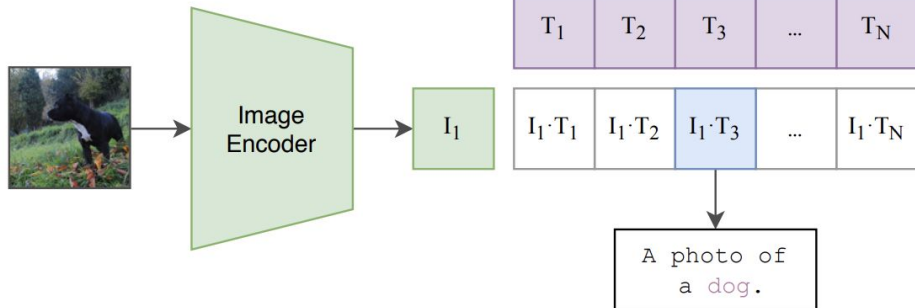


OpenAI's CLIP

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



```

# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss    = (loss_i + loss_t)/2

```

<https://arxiv.org/pdf/2103.00020.pdf>

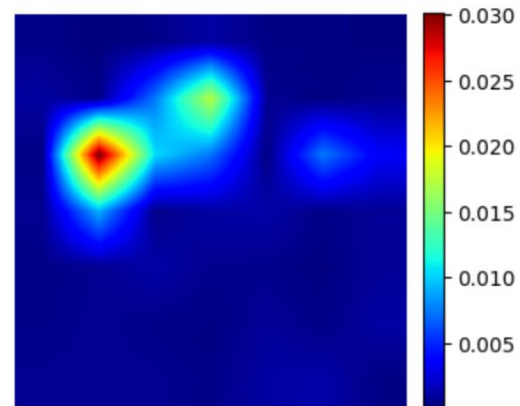
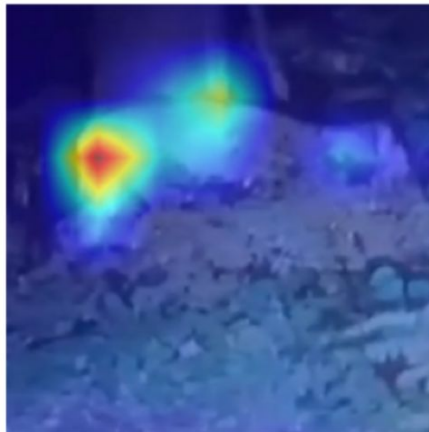
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

CLIP Explainability

Legend: ■ Negative □ Neutral ■ Positive

True Label Predicted Label Attribution Label Attribution Score Word Importance

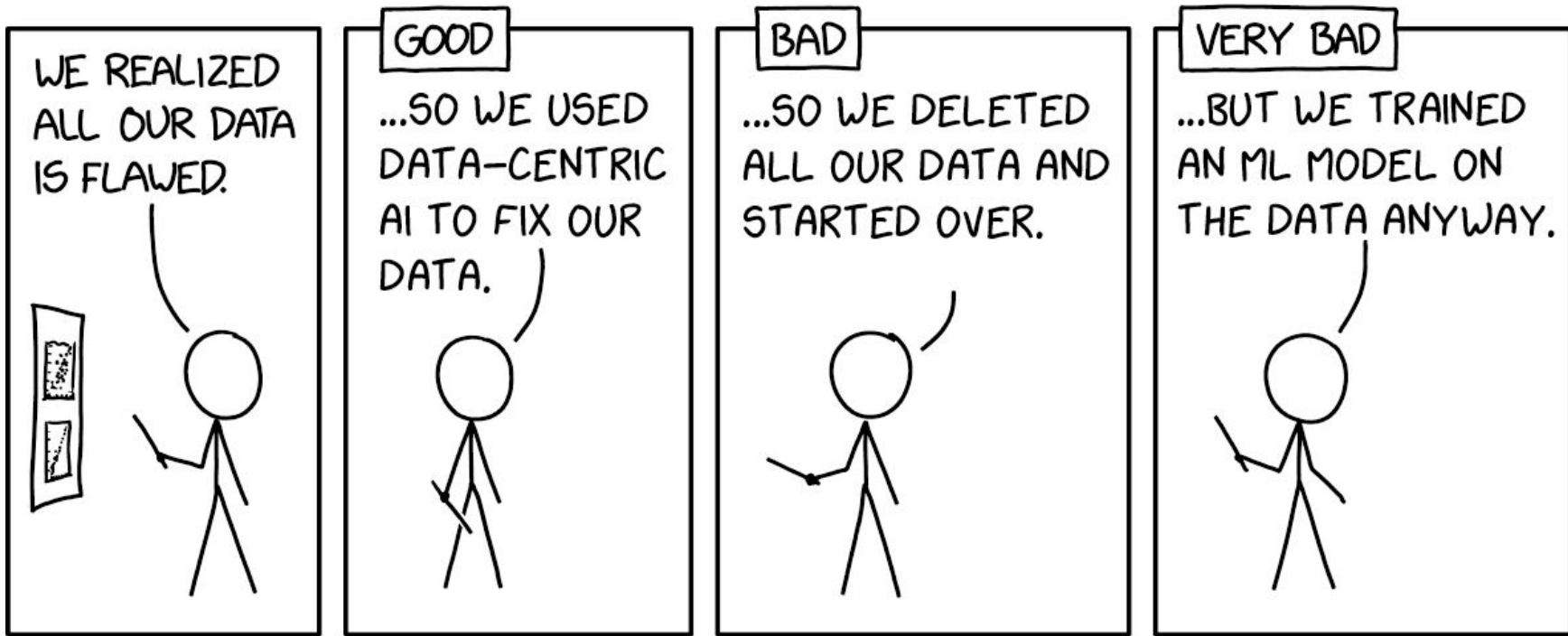
0 0 (0.00) 0 0.00 not a lion



Exploring CLIP

- [CLIP notebook](#)
- [CLIP explainability](#) (requires GPU credits to run)

Data-Centric AI



Labeling errors



ImageNet given label:

tick

Cleanlab guessed: **yellow garden spider**

MTurk consensus: **yellow garden spider**



ImageNet given label:

coyote

Cleanlab guessed: **dingo**

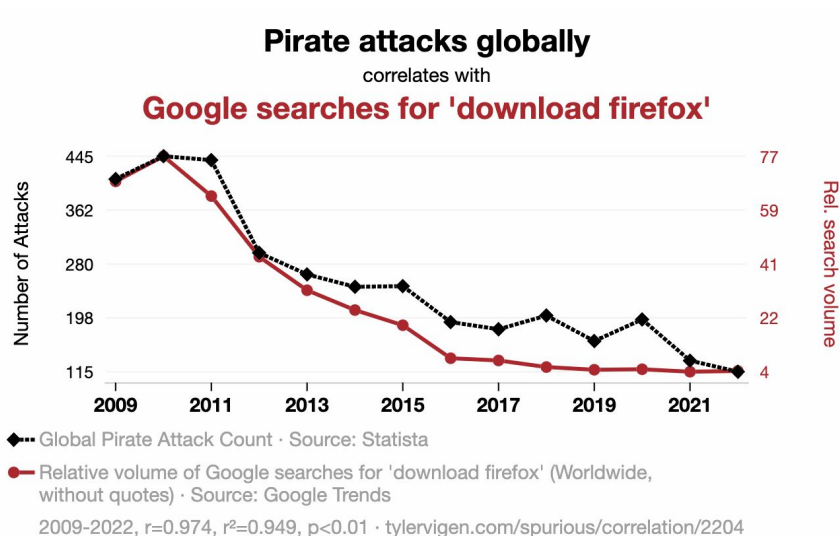
MTurk consensus: **dingo**

ID: 00012230

<https://labelerrors.com/>

The importance of clean training data

- Improves a model's accuracy and reliability
- Reduces overfitting to spurious correlations



[View details about correlation #2,204](https://www.tylervigen.com/spurious/correlation/2204)

<https://www.tylervigen.com/spurious-correlations>

Cleanlab's confident learning

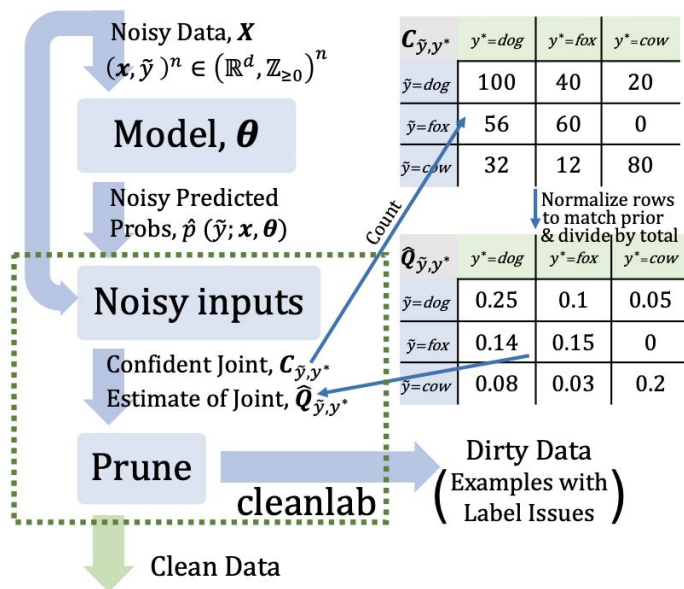


Figure 1: An example of the confident learning (CL) process. CL uses the confident joint, $C_{\tilde{y}, y^*}$, and $\hat{Q}_{\tilde{y}, y^*}$, an estimate of $Q_{\tilde{y}, y^*}$, the joint distribution of noisy observed labels \tilde{y} and unknown true labels y^* , to find examples with label errors and produce clean data for training.

Cleanlab and CLIP

id: 91
GL: boar
SL: cartoon



id: 37
GL: lion
SL: boar



id: 51
GL: boar
SL: cartoon



id: 76
GL: boar
SL: cartoon



id: 10
GL: lion
SL: boar



id: 17
GL: lion
SL: boar



id: 26
GL: lion
SL: boar



id: 32
GL: lion
SL: boar



id: 61
GL: boar
SL: cartoon



id: 56
GL: boar
SL: cartoon



id: 2
GL: lion
SL: boar



id: 45
GL: lion
SL: boar



id: 34
GL: lion
SL: boar



id: 40
GL: lion
SL: boar



id: 20
GL: lion
SL: boar



Imagelab

[Imagelab - a tool to identify near duplicates and low quality images](#)

Review questions

- How does the class-activation mapping work?
- What is zero-shot classification?
- How is CLIP trained?
- Can you explain the differences between train, validation, and test sets? What are the issues of having duplicates between these sets?
- Why are the performance metrics obtained on the dirty dataset **useless**?
- Why do we need to use cross-validation to clean the dataset on cleanlab?
What is hierarchical cross-validation?
- *Was it a lion?* 🤔

Join us for the next workshops!

- [AI Service Center - Berlin Brandenburg](#)

Next topics:

- **Meta's Segment Anything**
- **Segmentation and object detection with Detectron2 and YOLOv8**
- **Using Qdrant as a vector database for images and text**
- **Deployment of a computer vision system with FastAPI and Docker**